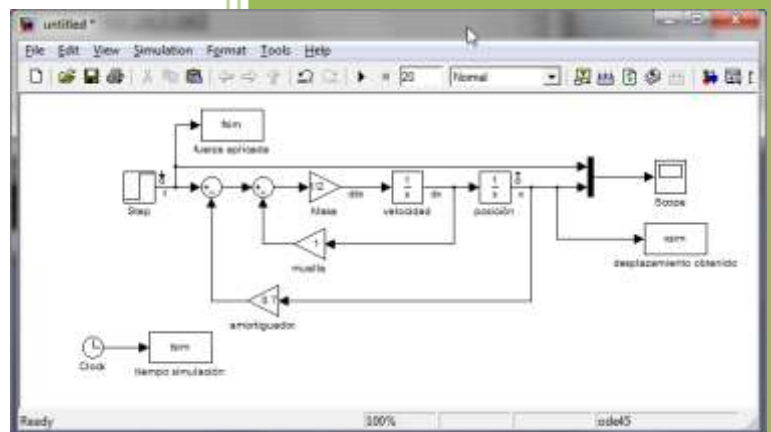
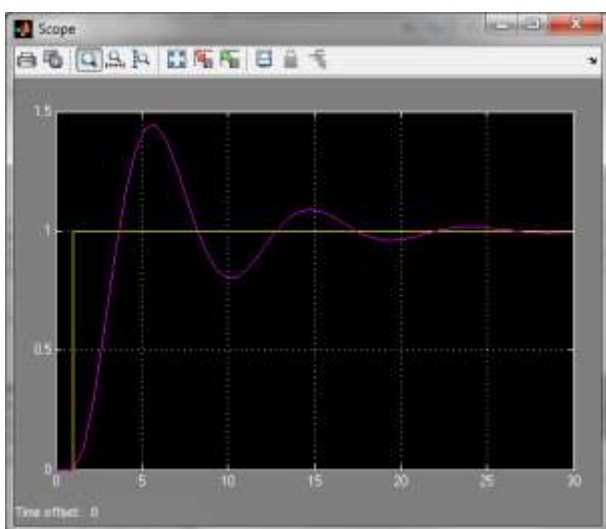
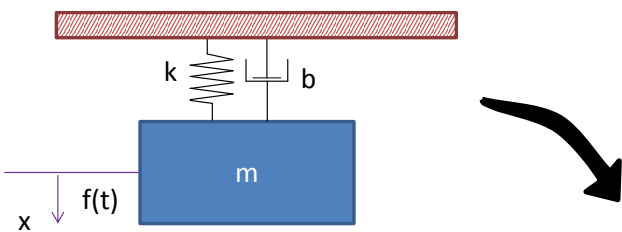


## Introducción al modelado y análisis de Sistemas con SIMULINK



## 0 Contenido

0	Contenido.....	1
1	MATLAB® y SIMULINK®.....	1
2	Uso de SIMULINK.....	1
2.1	Inicio del software.....	1
2.2	Creación de un modelo.....	3
2.2.1	Abrir un modelo existente.....	3
2.2.2	Introducción de bloques.....	4
2.2.3	Simulación.....	8
2.3	Simulación directa de la función de transferencia del modelo.....	11
2.4	Simulación de la función de transferencia del modelo con MATLAB.....	12

## 1 MATLAB® y SIMULINK®

MATLAB (MATrix LABoratory) es un lenguaje de alto nivel y un entorno interactivo para cálculo numérico, visualización y programación. Permite al ingeniero analizar datos, desarrollar algoritmos y crear modelos y aplicaciones propias. Sus puntos fundamentales son:

- Entorno interactivo para la exploración, diseño y solución de problemas.
- Amplia librería de funciones para cálculo científico que permiten trabajar de forma natural con problemas de algebra lineal, estadística, análisis de Fourier o cálculo numérico.
- Biblioteca de funciones gráficas para la visualización en 2D y 3D de datos. Incluye herramientas que permiten la generación de gráficos personalizados.
- Lenguaje de programación de alto nivel.
- Una gran variedad de librerías adicionales (toolboxes) especializadas en campos específicos de la ingeniería y la ciencia.

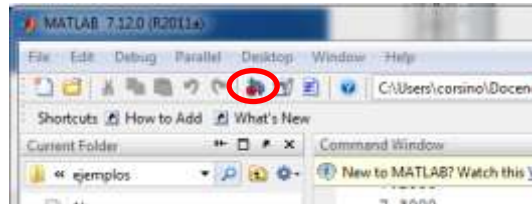
SIMULINK es una herramienta incluida en MATLAB para el modelado, simulación y análisis de sistemas dinámicos. El interfaz principal es una herramienta gráfica para la descripción del sistema mediante diagramas de bloques. Además, permite su extensión mediante la creación de librerías con nuevos tipos de bloques. Está altamente integrado con el resto de MATLAB. Es una herramienta muy usada en ingeniería de control y procesamiento de señal para la simulación de sistemas y el diseño basado en modelos.

## 2 Uso de SIMULINK

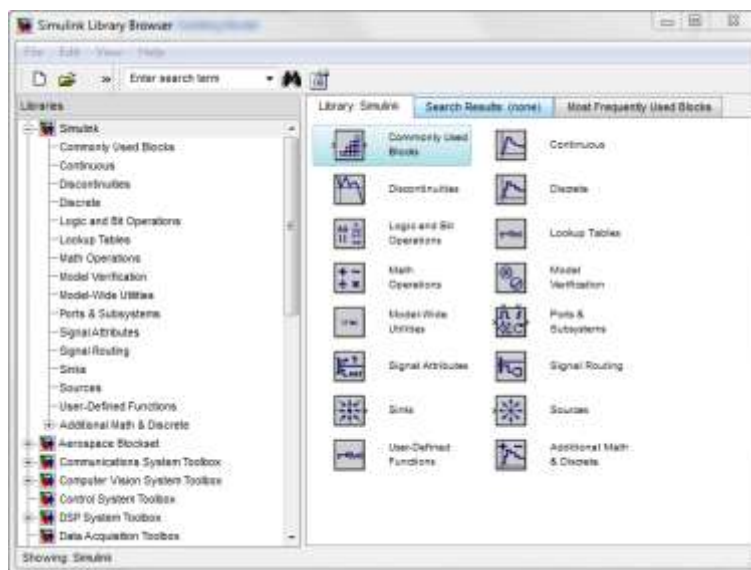
### 2.1 Inicio del software

El primer paso consiste en cargar el “Simulink Library Browser”, para lo que es necesario estar ejecutando MATLAB previamente. Una vez arrancado MATLAB, podemos:

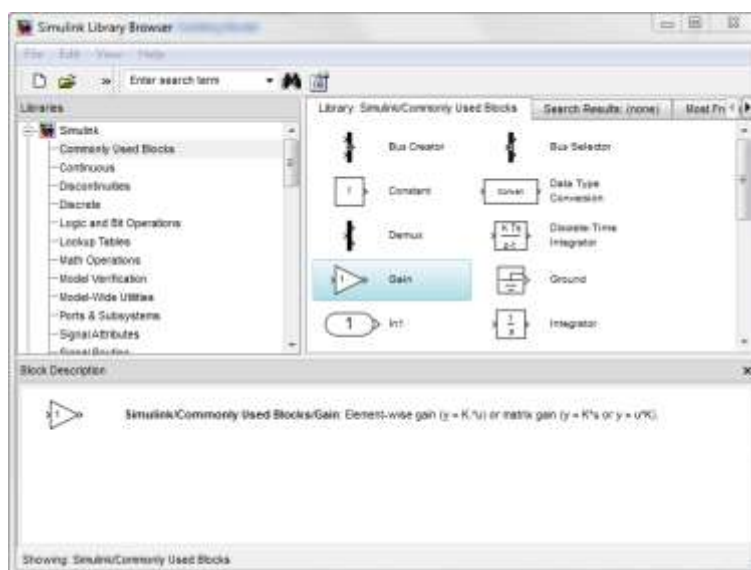
1. Ejecutar el comando *simulink* desde la línea de comandos de MATLAB.
2. Pulsar el botón “Simulink Library” que está en la barra de herramientas de MATLAB



Una vez cargado, aparecerá una ventana del Browser. Si es la primera vez que lo hacemos dentro de la sesión actual de MATLAB, la ventana puede tardar unos segundos en aparecer.



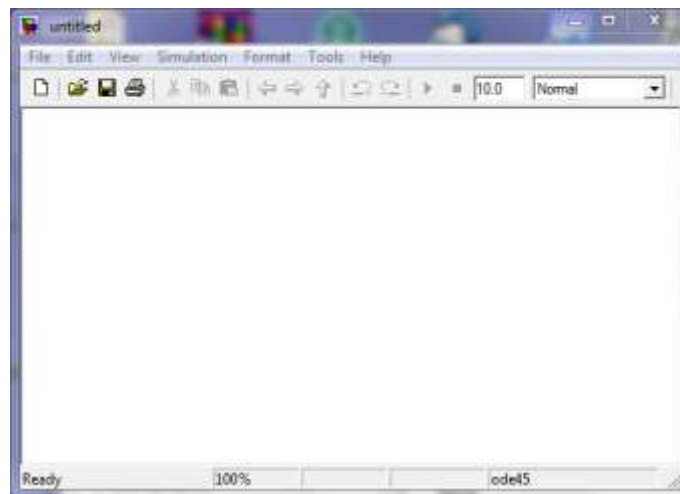
Si queremos que esta ventana esté siempre visible, podemos seleccionar la opción: *View > Stay on Top*. También podemos activar la opción *View > Show Block Descriptions* para abrir un panel en el que se visualiza una descripción del bloque seleccionado.



## 2.2 Creación de un modelo

Para crear un modelo propiamente dicho, debemos abrir una ventana del editor de SIMULINK:

1. En el menú del Simulink Library Browser o en la ventana principal de MATLAB, seleccionamos **File > New > Model**, lo que abrirá un modelo vacío. También es posible crear un nuevo modelo pulsando el icono “new file” en la barra de herramientas del Simulink Library Browser.
2. Para salvar el modelo, seleccionamos **File > Save** o **File > Save As** en el editor de Simulink. Si todavía no le hemos dado un nombre al modelo, siempre nos aparecerá el cuadro de diálogo Save As para introducir el nombre del fichero y seleccionar el directorio en el que queremos guardarlo.



### 2.2.1 Abrir un modelo existente

Los ficheros que contienen un modelo de Simulink se guardan con la extensión **.mdl**. Para abrir un modelo existente podemos:

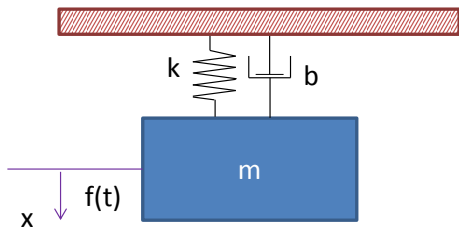
1. Seleccionar la opción **File > Open** en el Simulink Library Browser o en la ventana principal de MATLAB. Navegar hasta el directorio que contiene el archivo y seleccionarlo.
2. Desde la línea de comandos de MATLAB, escribir el nombre del archivo que contiene el modelo sin incluir la extensión. Por ejemplo para abrir el modelo ejemplo.mdl, deberíamos escribir ejemplo. Es necesario asegurarse previamente que el directorio que contiene el modelo es el directorio de trabajo o está incluido en el path de búsqueda de MATLAB. Para cambiar el directorio de trabajo se puede:
  - a. pulsar la barra de navegación incluida en la barra de herramientas de MATLAB o la que aparece en el panel “Current Folder”,



- b. utilizar el comando **cd** desde la línea de comandos de MATLAB.

### 2.2.2 Introducción de bloques

Consideremos el sistema formado por la amortiguación de un vehículo. Dicho sistema puede modelarse mediante un conjunto masa-muelle-amortiguador como el representado en la figura:



$$\ddot{x} = \frac{1}{m}(f(t) - b\dot{x} - kx)$$

$$G(s) = \frac{X(s)}{F(s)} = \frac{1}{2s^2 + bs + k}$$

Usaremos como parámetros del sistema:  $m=2\text{Kg}$ ,  $b=0.7\text{Ns/m}$  y  $K=1\text{N/m}$ .

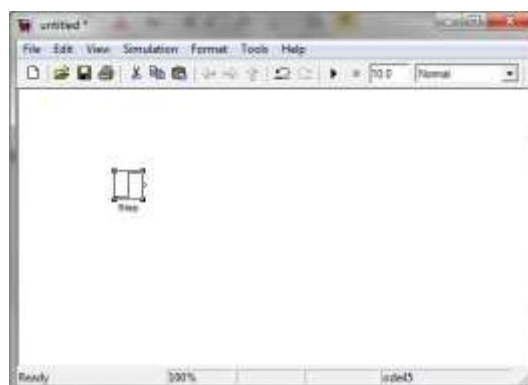
#### Introducción del modelo utilizando las ecuaciones

En primer lugar crearemos una nueva ventana para el modelo. Los bloques que necesitaremos son:

Bloques necesarios	Localización en el Simulink Library Browser
Step	Sources
Gain	Math Operation
Sum	Math Operation
Integrator	Continuous
Scope	Sinks
To Workspace	Sinks
Mux	Signal Routing
Clock	Sources

Para introducir las ecuaciones, debemos tener en cuenta que la posición ( $x$ ) es la integral de la velocidad ( $\dot{x}$ ) y que la velocidad es la integral de la aceleración ( $\ddot{x}$ ). También debemos tener en cuenta que la entrada a nuestro sistema es la fuerza  $f(t)$  y la salida la posición  $x(t)$ .

Para generar la fuerza en nuestro modelo, usaremos el bloque step que simula un cambio de tipo escalón en el valor de la fuerza aplicada.

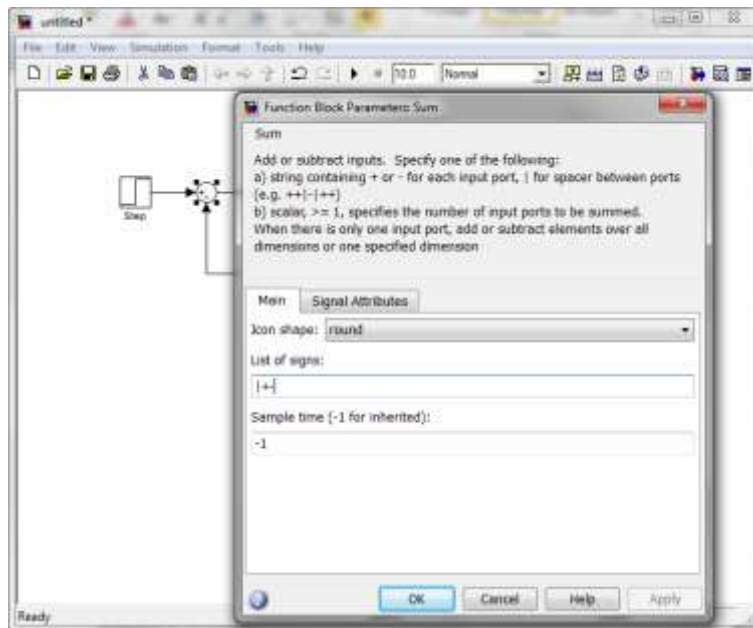


La derivada de mayor orden que aparece es la aceleración, que se obtiene:

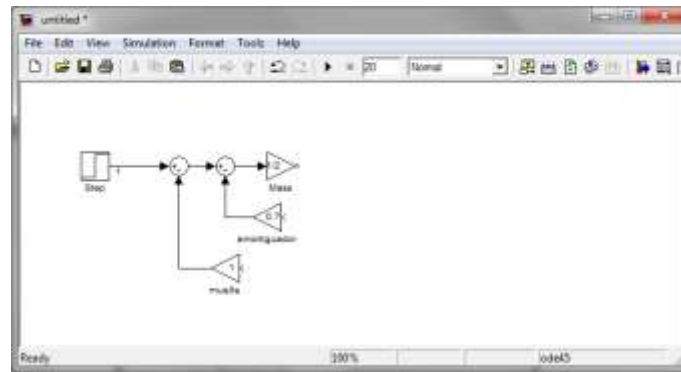
1. restando a la fuerza aplicada las fuerzas generadas por el muelle (proporcional a la posición) y por el amortiguador (proporcional a la velocidad).

2. Dividiendo el balance anterior por la masa del bloque.

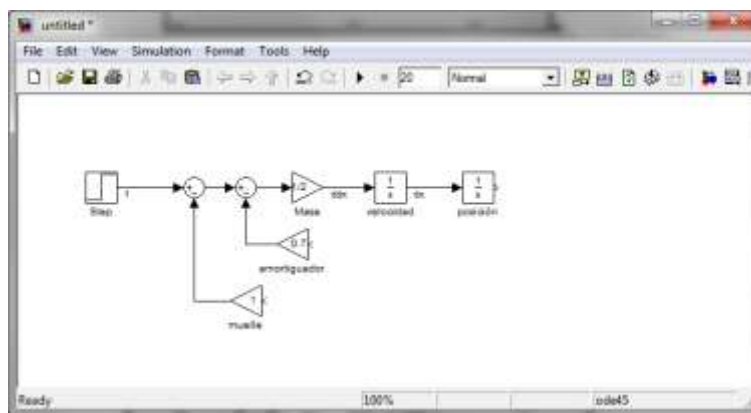
Para incluir un bloque sumador con una entrada positiva y una negativa, arrastraremos el bloque sum al modelo y haremos una doble pulsación sobre el elemento para abrir la ventana de parámetros del bloque. En el cuadro List of signs, sustuiremos la cadena “|++” por “|+-” para indicar que la primera posición no se usa, la segunda es una entrada positiva y la tercera una entrada inversora. Como tenemos que hacer un balance de tres cantidades, usaremos dos bloques sum.



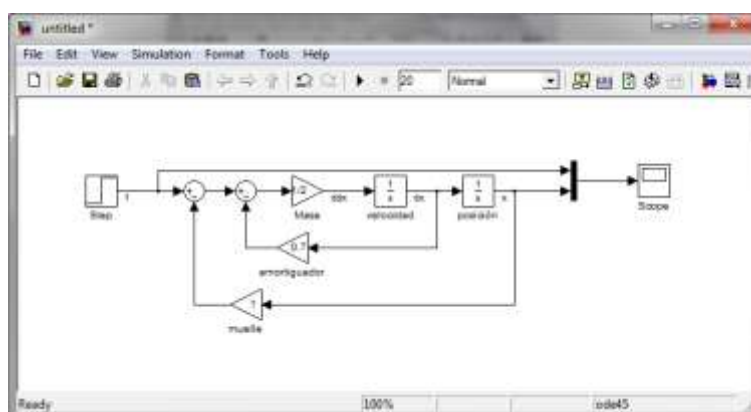
En el primer bloque sumador introduciremos la fuerza debida al muelle. Para ello, arrastramos un bloque de tipo gain al modelo. Para cambiar su orientación podemos seleccionar el bloque y pulsar **Ctrl+I** o usar la opción de menú **Format > Flip Block**. Por defecto el valor del bloque de ganancia es 1, que coincide con el valor de k en este ejemplo. Podemos cambiar el nombre del bloque pulsando dos veces sobre el mismo e introduciendo el nuevo texto, por ejemplo “muelle”. De la misma forma, introduciremos otro bloque ganancia para modelar el amortiguador. Para modificar la ganancia del bloque pulsaremos dos veces sobre el mismo y el cuadro de edición gain escribiremos 0.7. La salida del bloque que representa el amortiguador la conectaremos a la entrada inversora del segundo sumador. A la salida del segundo sumador tenemos ahora la fuerza resultante que actúa sobre la masa, si la dividimos por la masa del bloque (usando un bloque gain con ganancia 1/m) tendremos la aceleración a la que está sometido el bloque.



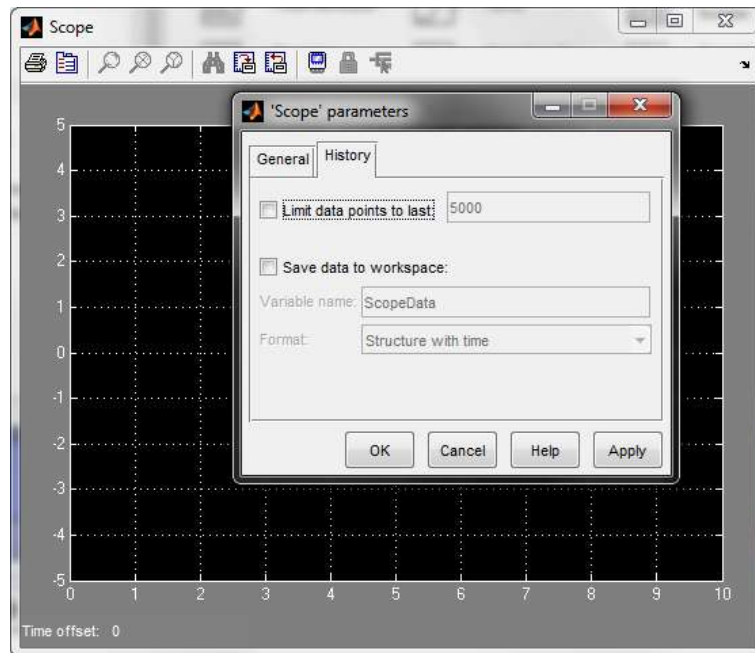
A partir de la aceleración, podemos obtener la velocidad por integración. Para ello, arrastramos un bloque de tipo Integrator al modelo. De la misma manera, podemos obtener la posición a partir de la velocidad.



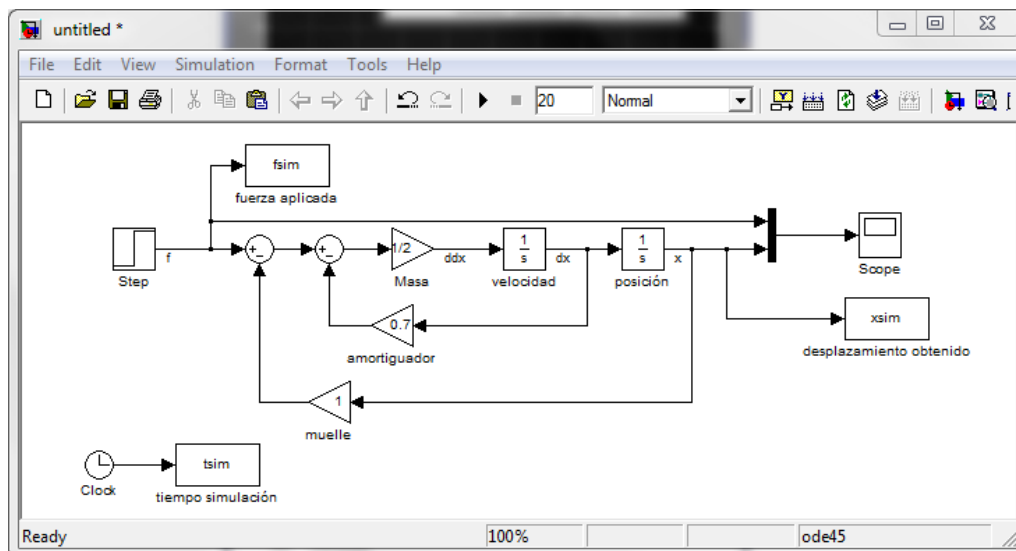
Ahora ya podemos llevar a las entradas de los bloques que representan el muelle y el amortiguador las señales correspondientes (posición y velocidad). Si queremos nombrar las señales basta con hacer una doble pulsación del botón izquierdo del ratón sobre la flecha que representa la señal y escribir el nombre en el recuadro que aparece. Para completar el modelo, incluiremos un osciloscopio en el que representaremos la fuerza aplicada y la posición de salida. Para ello incluiremos un bloque de tipo scope y otro de tipo mux en el modelo. El bloque mux es un bloque que multiplexa varias entradas en una única salida. A la primera entrada del bloque mux (la superior) conectaremos la fuerza aplicada y la segunda el desplazamiento obtenido. La salida del multiplexador se conecta a la entrada del osciloscopio.



Para configurar el osciloscopio, pulsaremos dos veces sobre él con el botón izquierdo del ratón. En la ventana que aparece, pulsaremos el icono parámetros (a la derecha del icono de impresora) y en la ventana de configuración que aparece seleccionamos la pestaña history. Desactivaremos la opción que limita los puntos a los últimos 5000. De este modo podremos ver toda la simulación.

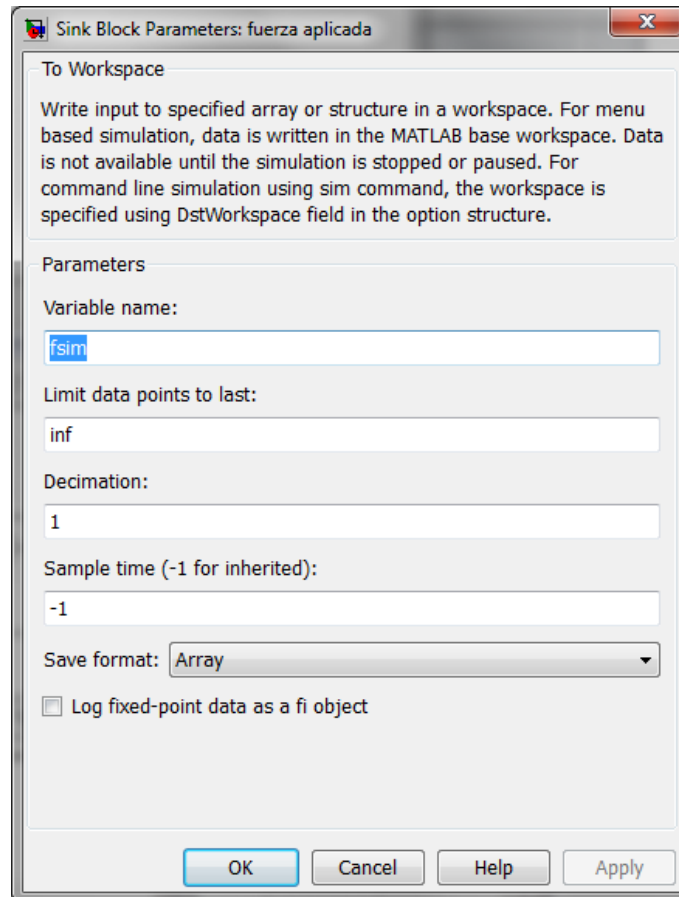


Por último, incluiremos bloques del tipo ToWorkspace con el fin de salvar en el espacio de trabajo de Matlab los resultados de la simulación. Para poder salvar en una variable los tiempos correspondientes a cada muestra de la simulación también es necesario incluir un bloque de tipo clock. El aspecto final del modelo es:



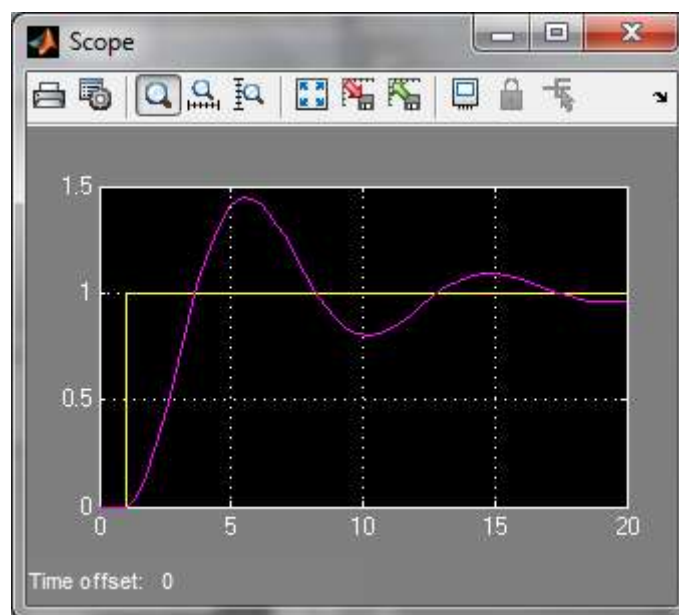
La configuración de los bloques ToWorkspace es como la que se muestra en la siguiente figura. En variable name pondremos el nombre de la variable en la que queremos guardar el resultado (por ejemplo fsim) y en el apartado Save Format elegiremos Array.



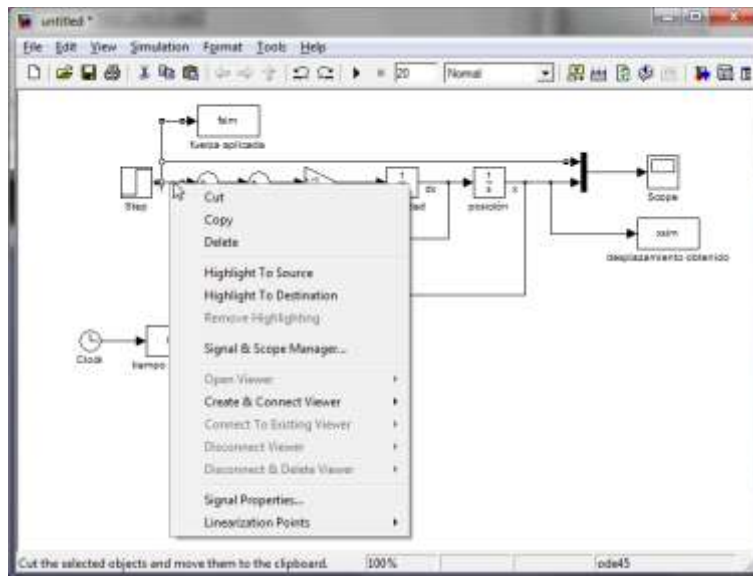


### 2.2.3 Simulación

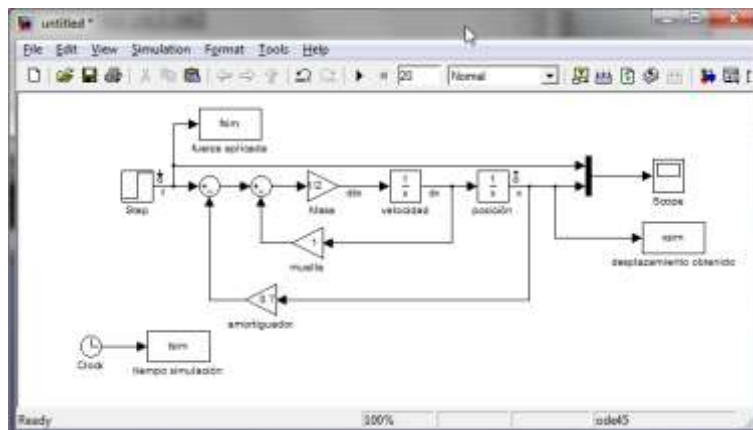
Para simular el modelo, ajustaremos el tiempo de la simulación a un valor adecuado a la dinámica de nuestro sistema (en este caso elegimos 20s) y pulsamos el botón play en la barra de herramientas del modelo. El resultado es el mostrado en la figura:



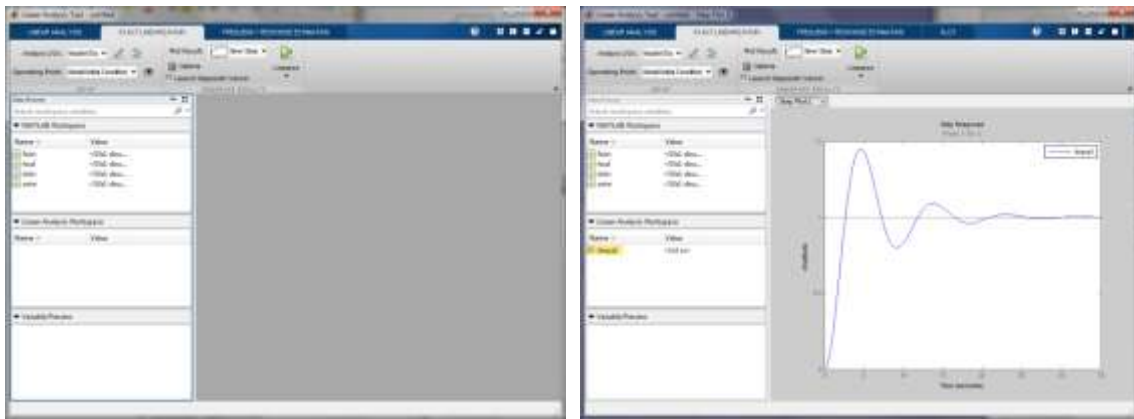
Podemos obtener un modelo lineal equivalente entre dos puntos del sistema. Para ello debemos marcar las entradas y salidas que nos interesan en nuestro modelo. Para fijar como entrada la señal  $f$ , pulsamos el botón derecho del ratón sobre la señal.



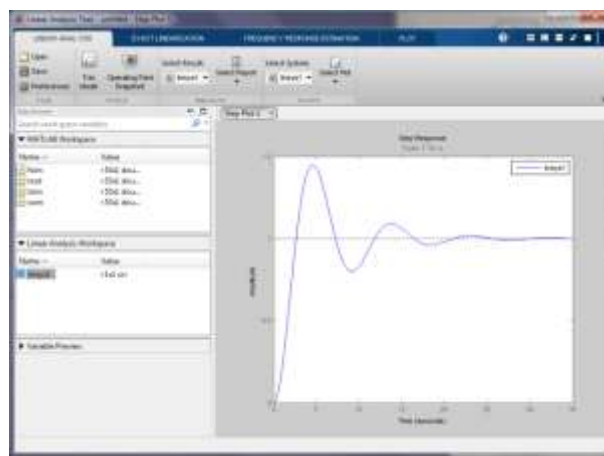
En la opción **Linearization Points**, seleccionaremos **input point**. Para marcar el desplazamiento como salida, seguiremos los mismos pasos, pero esta vez elegiremos la opción **output point**.



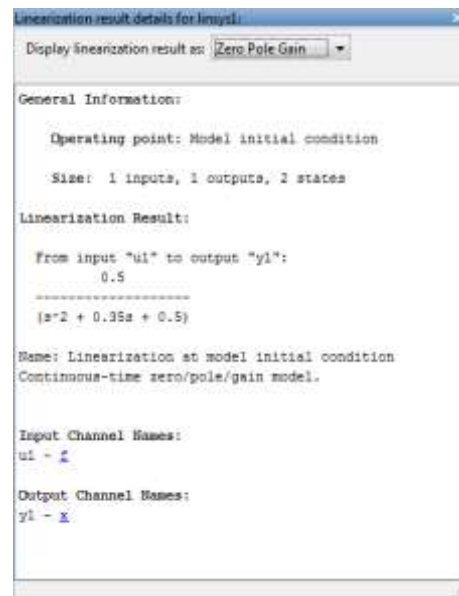
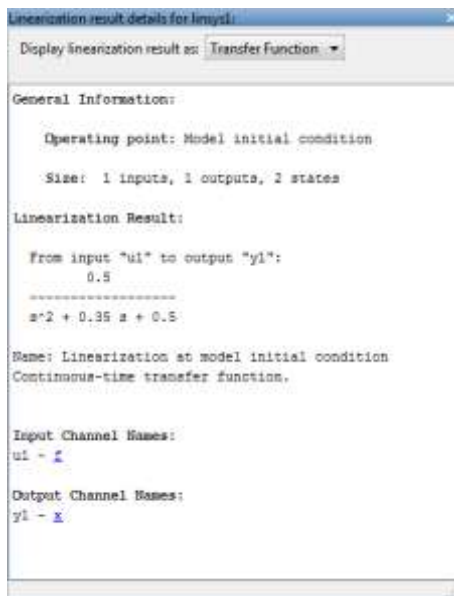
El siguiente paso es utilizar la herramienta **Tools > Control Design > Linear Analysis**. pulsaremos el botón Linearize. Nos aparecerá una ventana con la respuesta al escalón unitario del sistema linealizado.



Para ver la función de transferencia del modelo, seleccionamos la pestaña Linear Analysis:

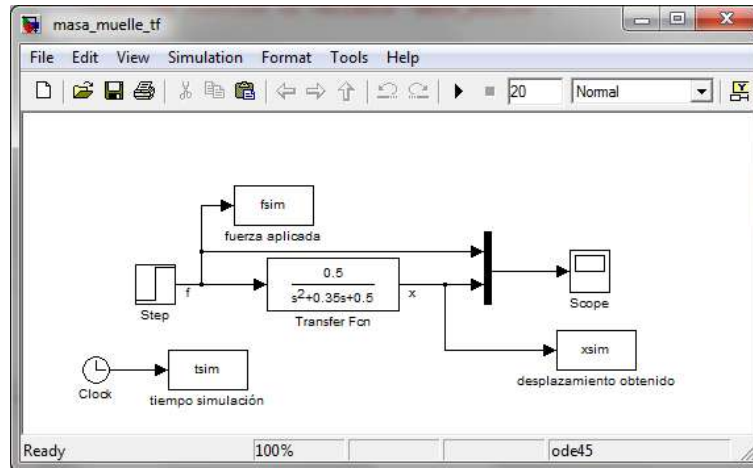


Seleccionamos la opción select report del bloque Results y en la ventana que aparece indicamos que nos muestre el resultado en formato Transfer Function o Zero Pole Gain:



### 2.3 Simulación directa de la función de transferencia del modelo

Si hubiésemos obtenido directamente la función de transferencia, podríamos simularla directamente usando un bloque de tipo Transfer Fcn que encontraremos en el grupo Continuous. El modelo que obtendríamos sería como el mostrado en la figura:



Para configurar el bloque con la función de transferencia, es suficiente con introducir dos vectores con los coeficientes del numerador y denominador de la función de transferencia.



## 2.4 Simulación de la función de transferencia del modelo con MATLAB

Por último se verá cómo realizar una simulación del modelo lineal, a partir de la función de transferencia, en la línea de comandos de MATLAB.

Para ello, lo primero que habrá que hacer es definir la función de transferencia:

```
>>num = 1; den = [2 1 1]; %se rellenan dos variables con los  
coeficientes del numerador y denominador de la fdt  
  
>>masamuelle = tf(num,den); %crea una variable de tipo sistema  
  
>>step(masamuelle); %lanza una simulación de la respuesta ante escalón
```

Se obtendrá como resultado la siguiente figura:

